

The parallel versus branching recurrences in computability logic*

Wenyan Xu and Sanyang Liu

Abstract

This paper shows that the basic logic induced by the parallel recurrence λ of Computability Logic (i.e., the one in the signature $\{\neg, \wedge, \vee, \lambda, \mathbb{Y}\}$) is a proper superset of the basic logic induced by the branching recurrence \circ (i.e., the one in the signature $\{\neg, \wedge, \vee, \circ, \mathbb{Q}\}$). The latter is known to be precisely captured by the cirquent calculus system **CL15**, conjectured by Japaridze to remain sound—but not complete—with λ instead of \circ . The present result is obtained by positively verifying that conjecture. A secondary result of the paper is showing that λ is strictly weaker than \circ in the sense that, while $\circ F$ logically implies λF , vice versa does not hold.

MSC: primary: 03B47; secondary: 03B70; 68Q10; 68T27; 68T15.

Keywords: Computability logic; Cirquent calculus; Interactive computation; Game semantics; Resource semantics.

1 Introduction

Computability logic (CoL), introduced by G. Japaridze [1, 5], is a formal theory of interactive computational problems, understood as games between a machine and its environment (symbolically named as \top and \perp , respectively). Formulas in it represent such problems; logical operators stand for operations on them; “truth” means existence of an algorithmic solution, i.e. \top ’s effective winning strategy; and validity is understood as truth under every particular interpretation of atoms.

Among the most important operators of CoL are *recurrence operators*, in their overall logical spirit reminiscent of the exponentials of linear logic. Recurrences, in turn, come in several flavors, two most natural and basic sorts of which are *parallel recurrence* λ and *branching recurrence* \circ , together with their duals \mathbb{Y}, \mathbb{Q} defined by $\mathbb{Y}F = \neg\lambda\neg F$ and $\mathbb{Q}F = \neg\circ\neg F$. Ample intuitive discussions and elaborations on the two sorts of recurrences and the relations between them were given in [3, 6, 8, 12]. However, finding syntactic characterizations of the logic induced by recurrences had been among the greatest challenges in CoL until the recent work [10, 11], where a sound and complete axiomatization, called **CL15**, for the basic $(\neg, \wedge, \vee, \circ, \mathbb{Q})$ -fragment of computability logic was constructed.¹ At the same time, the logical behavior of parallel recurrence λ still remains largely ununderstood. It is not even known whether the set of principles validated by λ is recursively enumerable. The present paper brings some initial light into this otherwise completely dark picture. It shows that the set of principles validated by λ, \mathbb{Y} in combination with the basic operations \neg, \wedge, \vee is a proper superset of the set of those validated by \circ, \mathbb{Q} . This is achieved by positively settling Conjecture 6.3 of [10], according to which **CL15** continues to be sound—but not complete—with λ and \mathbb{Y} instead of \circ and \mathbb{Q} . Further, to make our investigation of the relationship between λ and \circ more complete, at the end of the paper we also prove that λ is strictly weaker than \circ in the sense that, while $\circ F$ logically implies λF (as shown in [3]), vice versa does not hold.

CL15 is a system built in *cirquent calculus*. The latter is a refinement of sequent calculus. Unlike the more traditional proof theories that manipulate tree-like objects (formulas, sequents, hypersequents, etc.), cirquent calculus deals with graph-style structures called *cirquents* (the term is a combination of “CIRcuit” and “seQUENT”), with its main characteristic feature being allowing to explicitly account

*Supported by the NNSF (60974082) of China.

¹The soundness part was proven in [10], and the completeness part in [11].

for possible *sharing* of subcomponents between different subcomponents. The approach was introduced by Japaridze in [2] as a new deductive tool for CoL and was further developed in [4, 7, 13, 14] where a number of advantages of this novel sort of proof theory were revealed, such as high expressiveness, flexibility and efficiency.

In order to make this paper reasonably self-contained, in the next section we reproduce the basic concepts from [5, 10] on which the later parts of the paper will rely. An interested reader may consult [5, 10] for the associated motivations, detailed explanations and examples.

2 Preliminaries

The letter \wp is used as a variable ranging over $\{\top, \perp\}$, with $\neg\wp$ meaning \wp 's adversary. A **move** is a finite string over standard keyboard alphabet. A **labmove** is a move prefixed (“labeled”) with \top or \perp . A **run** is a finite or infinite sequence of labmoves, and a **position** is a finite run. Runs are usually delimited by “⟨” and “⟩”, with ⟨⟩ thus denoting the **empty run**. For any run Γ , $\neg\Gamma$ is the same as Γ , with the only difference that every label \wp is changed to $\neg\wp$.

A **game**² is a pair $A = (\mathbf{Lr}^A, \mathbf{Wn}^A)$, where: (1) \mathbf{Lr}^A is a set of runs satisfying the condition that a finite or infinite run Γ is in \mathbf{Lr}^A iff so are all of Γ 's nonempty finite initial segments.³ If $\Gamma \in \mathbf{Lr}^A$, then Γ is said to be a **legal run** of A ; otherwise Γ is an **illegal run** of A . A move α is a **legal move** for a player \wp in a position Φ of A iff $\langle \Phi, \wp\alpha \rangle \in \mathbf{Lr}^A$; otherwise α is an **illegal move**. When the last move of the shortest illegal initial segment of Γ is \wp -labeled, Γ is said to be a \wp -**illegal run** of A . (2) \mathbf{Wn}^A is a function that sends every run Γ to one of the players \top or \perp , satisfying the condition that if Γ is a \wp -illegal run of A , then $\mathbf{Wn}^A(\Gamma) = \neg\wp$. When $\mathbf{Wn}^A(\Gamma) = \wp$, Γ is said to be a \wp -**won** run of A .

The game operations dealt with in the present paper are \neg (negation), \vee (parallel disjunction), \wedge (parallel conjunction), λ (parallel recurrence), γ (parallel corecurrence), \circ (branching recurrence) and \wp (branching corecurrence). Intuitively, \neg is a role switch operator: $\neg A$ is the game A with the roles of \top and \perp interchanged (\top 's legal moves and wins become those of \perp , and vice versa). Both $A \wedge B$ and $A \vee B$ are games playing which means playing the two components A and B simultaneously (in parallel). In $A \wedge B$, \top is the winner if it wins in both components, while in $A \vee B$ winning in just one component is sufficient. Next, λA is nothing but the infinite parallel conjunction $A \wedge A \wedge A \wedge \dots$, and γA is nothing but the infinite parallel disjunction $A \vee A \vee A \vee \dots$. Finally, both $\circ A$ and $\wp A$ are games playing which means simultaneously playing a continuum of copies (or “threads”) of A . Each copy/thread is denoted by an infinite bitstring and vice versa, where a **bitstring** is a finite or infinite sequence of bits 0,1. Making a move $w.\alpha$, where w is a finite bitstring, means making the move α simultaneously in all threads of the form wy . In $\circ A$, \top is the winner iff it wins in all threads of A , while in $\wp A$ winning in just one thread is sufficient. Again, it should be pointed out that the above is just a very brief and incomplete intuitive characterization. See [5] for more.

Let Γ be a run and α be a move. The notation

$$\Gamma^\alpha$$

will be used to indicate the result of deleting from Γ all moves (together with their labels) except those that look like $\alpha\beta$ for some move β , and then further deleting the prefix “ α ” from such moves. For instance, $\langle \top 1.\alpha, \perp 2.\beta, \top 1.\gamma, \perp 2.\delta \rangle^1 = \langle \top \alpha, \top \gamma \rangle$.

Let Ω be a run and x be an infinite bitstring. The notation

$$\Omega^{\preceq x}$$

will be used to indicate the result of deleting from Ω all moves (together with their labels) except those that look like $u.\beta$ for some move β and some finite initial segment u of x , and then further deleting the prefix “ $u.$ ” from such moves. For instance, $\langle \perp 10.\alpha, \top 111.\beta, \perp 1.\gamma, \perp 00.\alpha \rangle^{\preceq 111\dots} = \langle \top \beta, \perp \gamma \rangle$.

²The concept of a game considered in CoL is more general than the one defined here, with games in the present sense called *constant games*. Since we (for simplicity) only consider constant games in this paper, we omit the word “constant” and just say “game”.

³This condition can be seen to imply that the empty run ⟨⟩ is always in \mathbf{Lr}^A .

The earlier-outlined intuitive characterizations of the game operators are captured by the following formal definition. Below, A , A_1 , A_2 are arbitrary games, α ranges over moves, $i \in \{1, 2\}$, u ranges over positive integers identified with its decimal representation, w ranges over finite bitstrings, x ranges over infinite bitstrings, Γ is an arbitrary run, and Ω is any legal run of the game that is being defined.

1. $\neg A$ (**negation**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{\neg A}$ iff $\neg\Gamma \in \mathbf{Lr}^A$.
- (ii) $\mathbf{Wn}^{\neg A}\langle\Omega\rangle = \top$ iff $\mathbf{Wn}^A\langle\neg\Omega\rangle = \perp$.

2. $A_1 \wedge A_2$ (**parallel conjunction**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{A_1 \wedge A_2}$ iff every move of Γ is $i.\alpha$ for some i, α and, for both i , $\Gamma^{i.} \in \mathbf{Lr}^{A_i}$.
- (ii) $\mathbf{Wn}^{A_1 \wedge A_2}\langle\Omega\rangle = \top$ iff, for both i , $\mathbf{Wn}^{A_i}\langle\Omega^{i.}\rangle = \top$.

3. $A_1 \vee A_2$ (**parallel disjunction**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{A_1 \vee A_2}$ iff every move of Γ is $i.\alpha$ for some i, α and, for both i , $\Gamma^{i.} \in \mathbf{Lr}^{A_i}$.
- (ii) $\mathbf{Wn}^{A_1 \vee A_2}\langle\Omega\rangle = \top$ iff, for some i , $\mathbf{Wn}^{A_i}\langle\Omega^{i.}\rangle = \top$.

4. λA (**parallel recurrence**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{\lambda A}$ iff every move of Γ is $u.\alpha$ for some u and α and, for each such u , $\Gamma^{u.} \in \mathbf{Lr}^A$.
- (ii) $\mathbf{Wn}^{\lambda A}\langle\Omega\rangle = \top$ iff, for all u , $\mathbf{Wn}^A\langle\Omega^{u.}\rangle = \top$.

5. γA (**parallel corecurrence**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{\gamma A}$ iff every move of Γ is $u.\alpha$ for some u and α and, for each such u , $\Gamma^{u.} \in \mathbf{Lr}^A$.
- (ii) $\mathbf{Wn}^{\gamma A}\langle\Omega\rangle = \top$ iff, for some u , $\mathbf{Wn}^A\langle\Omega^{u.}\rangle = \top$.

6. δA (**branching recurrence**)⁴ is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{\delta A}$ iff every move of Γ is $w.\alpha$ for some w, α and, for all x , $\Gamma^{\preceq x} \in \mathbf{Lr}^A$.
- (ii) $\mathbf{Wn}^{\delta A}\langle\Omega\rangle = \top$ iff, for all x , $\mathbf{Wn}^A\langle\Omega^{\preceq x}\rangle = \top$.

7. ϑA (**branching corecurrence**) is defined by:

- (i) $\Gamma \in \mathbf{Lr}^{\vartheta A}$ iff every move of Γ is $w.\alpha$ for some w, α and, for all x , $\Gamma^{\preceq x} \in \mathbf{Lr}^A$.
- (ii) $\mathbf{Wn}^{\vartheta A}\langle\Omega\rangle = \top$ iff, for some x , $\mathbf{Wn}^A\langle\Omega^{\preceq x}\rangle = \top$.

In what follows, we explain—formally or informally—several additional concepts relevant to our proofs.

(1) **Static games:** CoL restricts its attention to a special yet very wide subclass of games termed “static”. Intuitively, static games are interactive tasks where the relative speeds of the players are irrelevant, as it never hurts a player to postpone making moves. A formal definition of this concept can be found in [5], which we will not reproduce here as nothing in this paper relies on it. The only relevant for us fact, proven in [1, 5, 9], is that the class of static games is closed under the operations $\neg, \wedge, \vee, \lambda, \gamma, \delta, \vartheta$ (as well as any other game operations studied in CoL).

(2) **EPM:** CoL understands \top ’s effective strategies as interactive machines. Several sorts of such machines have been proposed and studied in CoL, all of them turning out to be equivalent in computing power once we exclusively consider static games. In this paper we only use one sort of such machines, called the *easy-play machine* (**EPM**). It is a kind of a Turing machine with the additional capability of making moves, and has two tapes⁵: the ordinary read/write *work tape*, and the read-only *run tape*. The run tape serves as a dynamic input, at any time (“**clock cycle**”) spelling the current position: every time one of the players makes a move, that move—with the corresponding label—is automatically

⁴The present version of branching (co)recurrence was introduced recently in [9]. It is different from yet equivalent to (in all relevant respects) the older version found in [1, 5]. The same applies to ϑ .

⁵Often there is also a third tape called the *valuation tape*. Its function is to provide values for the variables on which a game may depend. However, as we remember, in this paper we only consider constant games — games that do not depend on any variables. This makes it possible to safely remove the valuation tape (or leave it there but fully ignore), as this tape is no longer relevant.

appended to the content of this tape. The machine can make a (one single) move at any time, while its environment can make an (at most one) move only when the machine explicitly allows it to do so (this sort of an action is called **granting permission**).⁶

(3) **Strategies:** Let \mathcal{M} be an EPM. A *configuration* of \mathcal{M} is a full description of the current state of the machine, the contents of its two tapes, and the locations of the corresponding two scanning heads. The *initial configuration* is the configuration where \mathcal{M} is in its start state and both tapes are empty. A configuration C' is said to be an *successor* of a configuration C if C' can legally follow C in the standard sense, based on the (deterministic) transition function of the machine and accounting for the possibility of nondeterministic updates of the content of the run tape through environment's moves. A **computation branch** of \mathcal{M} is a sequence of configurations of \mathcal{M} where the first configuration is the initial configuration, and each other configuration is a successor of the previous one. Each computation branch B of \mathcal{M} incrementally spells a run Γ on the run tape, which is called the **run spelled by** B . Subsequently, any such run Γ will be referred to as a **run generated by** \mathcal{M} . A computation branch B of \mathcal{M} is said to be **fair** iff, in it, permission has been granted infinitely many times. An **algorithmic solution** (\top 's winning strategy) for a given game A is understood as an EPM \mathcal{M} such that, whenever B is a computation branch of \mathcal{M} and Γ the run spelled by B , Γ is a \top -won run of A , where B should be fair unless Γ is a \perp -illegal run of A . When the above is the case, we say that \mathcal{M} **wins** A .

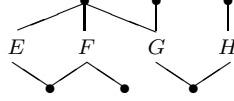
Now about formulas and the underlying semantics. We have some fixed set of syntactic objects, called **atoms**, for which P, Q, R will be used as metavariables. A **formula** is built from atoms in the standard way using the connectives $\neg, \vee, \wedge, \lambda, \gamma, \circ, \diamond$, with $F \rightarrow G$ understood as an abbreviation for $\neg F \vee G$ and \neg limited only to atoms, where $\neg\neg F$ is understood as F , $\neg(F \wedge G)$ as $\neg F \vee \neg G$, $\neg(F \vee G)$ as $\neg F \wedge \neg G$, $\neg\lambda F$ as $\gamma\neg F$, $\neg\gamma F$ as $\lambda\neg F$, $\neg\circ F$ as $\diamond\neg F$, and $\neg\diamond F$ as $\circ\neg F$. A $(\neg, \wedge, \vee, \lambda, \gamma)$ -**formula** is one not containing \circ, \diamond . Similarly, a $(\neg, \wedge, \vee, \circ, \diamond)$ -**formula** is one not containing λ, γ . An **interpretation** is a function $*$ that sends every atom P to a static game P^* , and extends to all formulas by seeing the logical connectives as the same-name game operations. A formula F is **uniformly valid** iff there is an EPM \mathcal{M} , called a **uniform solution** of F , such that, for every interpretation $*$, \mathcal{M} wins F^* .⁷ Throughout the rest of this paper, unless otherwise specified or suggested by the context, by a "formula" we will always mean a $(\neg, \wedge, \vee, \lambda, \gamma)$ -formula.

As noted in section 1, **CL15** is built in *cirquent calculus*, whose formalism goes beyond formulas. Namely, a **cirquent** is a triple $C = (\vec{F}, \vec{U}, \vec{O})$ where: (1) \vec{F} is a nonempty finite sequence of formulas, whose elements are said to be the **oformulas** of C . Here the prefix "o" is used to mean a formula together with a particular occurrence of it in \vec{F} . For instance, if $\vec{F} = \langle G, H, H \rangle$, then the cirquent has three oformulas while only two formulas. (2) Both \vec{U} and \vec{O} are nonempty finite sequences of nonempty sets of oformulas of C . The elements of \vec{U} are said to be the **undergroups** of C , and the elements of \vec{O} are said to be the **overgroups** of C . Again, two undergroups (resp. overgroups) may be identical as sets (have identical **contents**), yet they count as different undergroups (resp. overgroups) because they occur at different places in \vec{U} (resp. \vec{O}). (3) Additionally, every oformula is required to be in at least one undergroup and at least one overgroup.

Rather than writing cirquents as ordered tuples in the above style, we prefer to represent them through (and identify them with) **diagrams**. Below is such a representation for the cirquent that has four oformulas E, F, G, H , three undergroups $\{E, F\}$, $\{F\}$, $\{G, H\}$ and three overgroups $\{E, F, G\}$, $\{G\}$, $\{H\}$.

⁶In the more basic sort of machines called *hard-play machines* (**HPM**), the environment can make any number of moves at any time (needing no "permission" for that). It is known ([1, 5]) that the two sorts of machines win the same static games.

⁷Another sort of validity studied in CoL is multiform validity. A formula F is **multiformly valid** iff, for every interpretation $*$, there is a machine that wins F^* . Since uniform validity is stronger than multiform validity, all soundness-style results that we are going to establish about uniform validity automatically extend to multiform validity as well. Partly for this reason, in this paper we will be exclusively interested in uniform validity.



Each group in the diagram is represented by (and identified with) a \bullet , where the **arcs** (lines connecting the \bullet with oformulas) are pointing to the oformulas that the given group contains.

There are ten inference rules in **CL15**. Below we reproduce those rules from [10] with \circlearrowleft and \circlearrowright rewritten as λ and γ , respectively. To semantically differentiate the two versions of **CL15** (when necessary), we may use the name **CL15**(\circlearrowleft) for the system that understands (and writes) the recurrence operator as \circlearrowleft , and use **CL15**(λ) for the system that understands (and writes) the recurrence operator as λ .

Axiom (A): Axiom is a “rule” with no premises. It introduces the cirquent

$$((\neg F_1, F_1, \dots, \neg F_n, F_n), \langle \{\neg F_1, F_1\}, \dots, \{\neg F_n, F_n\} \rangle, \langle \{\neg F_1, F_1\}, \dots, \{\neg F_n, F_n\} \rangle),$$

where n is any positive integer, and F_1, \dots, F_n are any formulas. All rules other than Axiom take a single premise.

Exchange (E): This rule comes in three versions: **Undergroup Exchange**, **Oformula Exchange** and **Overgroup Exchange**. The conclusion of Oformula Exchange is obtained by interchanging in the premise two adjacent oformulas E and F , and redirecting to E (resp. F) all arcs that were originally pointing to E (resp. F). Undergroup (resp. Overgroup) Exchange is the same, with the only difference that the objects interchanged are undergroups (resp. overgroups).

Duplication (D): This rule comes in two versions: **Undergroup Duplication** and **Overgroup Duplication**. The conclusion of Undergroup Duplication is obtained by replacing in the premise some undergroup U with two adjacent undergroups whose contents are identical to that of U . Similarly for Overgroup Duplication.

Merging (M): The conclusion of this rule can be obtained from the premise by merging any two adjacent overgroups O_1 and O_2 into one overgroup O , and including in O all oformulas that were originally contained in O_1 or O_2 or both.

Weakening (W): For the convenience of description, we explain this and the remaining rules in the bottom-up view. The premise of this rule is obtained by deleting in the conclusion an arc between some undergroup U with ≥ 2 elements and some oformula F ; if U was the only undergroup containing F , then F should also be deleted, together with all arcs between F and overgroups; if such a deletion makes some overgroups empty, then they should also be deleted.

Contraction (C): The premise of this rule is obtained by replacing in the conclusion an oformula γF by two adjacent oformulas γF and γF , and including both of them in exactly the same undergroups and overgroups in which the original γF was contained.

Disjunction introduction (\vee): The premise of this rule is obtained by replacing in the conclusion an oformula $E \vee F$ by two adjacent oformulas E and F , and including both of them in exactly the same undergroups and overgroups in which the original $E \vee F$ was contained.

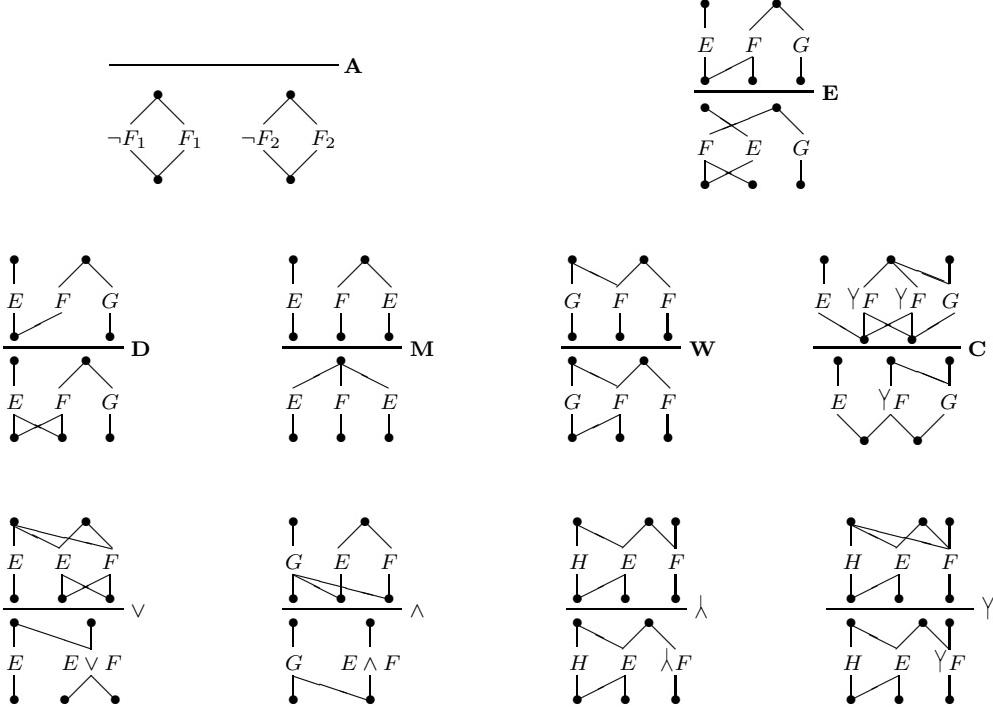
Conjunction introduction (\wedge): According to this rule, if a cirquent (the conclusion) has an oformula $E \wedge F$, then the premise can be obtained by splitting the original $E \wedge F$ into two adjacent oformulas E and F , including both of them in exactly the same overgroups in which the original $E \wedge F$ was contained, and splitting every undergroup Γ that originally contained $E \wedge F$ into two adjacent undergroups Γ^E and Γ^F , where Γ^E contains E (but not F), and Γ^F contains F (but not E), with all other ($\neq E \wedge F$) oformulas of Γ contained by both Γ^E and Γ^F .

Recurrence introduction (λ): The premise of this rule is obtained by replacing in the conclusion an oformula λF by F , with all arcs unchanged, and inserting a new overgroup Γ that contains F as its *only* oformula.

Corecurrence introduction (γ): The premise of this rule is obtained by replacing in the conclusion an oformula γF by F , with all arcs unchanged, and additionally including F in any (possibly zero) number of the already existing overgroups.

Below we provide illustrations for all rules, in each case an abbreviated name of the rule standing

next to the horizontal line separating the premise from the conclusion. Our illustration for the axiom (the “A” labeled rule) is a specific cirquent where $n = 2$; our illustrations for all other rules are merely examples chosen arbitrarily. Unfortunately, no systematic ways for schematically representing cirquent calculus rules have been elaborated so far. This explains why we appeal to examples instead.



The above are all ten rules of $\text{CL15}(\lambda)$. A $\text{CL15}(\lambda)$ -proof (or simply a proof) of a cirquent C is a sequence $\langle C_1, \dots, C_n \rangle$ of cirquents, where $n \geq 1$, such that $C_n = C$, C_1 is an axiom, and C_i ($1 < i \leq n$) follows from C_{i-1} by one of the rules of $\text{CL15}(\lambda)$. For any formula F , the expression F^\bullet is used to denote the cirquent $(\langle F \rangle, \langle \{F\} \rangle, \langle \{F\} \rangle)$. Then a $\text{CL15}(\lambda)$ -proof (or simply a proof) of a formula F is stipulated to be a proof of the cirquent F^\bullet . A formula or cirquent X is provable, symbolically $\text{CL15}(\lambda) \vdash X$, iff it has a proof.

As mentioned, $\text{CL15}(\Diamond)$ is the same as $\text{CL15}(\lambda)$, only with \Diamond, \Diamond instead of λ, γ .

Theorem 2.1 (Japaridze [10, 11]) *A $(\neg, \wedge, \vee, \Diamond, \Diamond)$ -formula is uniformly valid iff it is provable in $\text{CL15}(\Diamond)$.*

3 A semantics of cirquents

To prove the soundness of $\text{CL15}(\lambda)$, we need to extend the earlier-described semantics from formulas to cirquents.

Notation 3.1 Let Γ be a run, a be a positive integer, and $\vec{x} = x_1, \dots, x_n$ be a nonempty sequence of n positive integers. We will be using the notation

$$\Gamma^{[a; \vec{x}]}$$

to mean the result of

- deleting from Γ all moves (together with their labels) except those that look like $a; u_1, \dots, u_n. \beta$ for some move β and some sequence of n natural numbers u_1, \dots, u_n satisfying the condition that whenever $u_i \neq 0$ ($i \in \{1, \dots, n\}$), $u_i = x_i$, and

- then further deleting the prefix “ $a; u_1, \dots, u_n$.” from such moves.

For instance, $\langle \perp 1; 1, 1.\alpha, \top 1; 1, 2.\beta, \perp 1; 1, 0.\gamma, \perp 2; 1, 0.\delta \rangle^{[1;1,2]} = \langle \top \beta, \perp \gamma \rangle$.

Definition 3.2 Let $*$ be an interpretation, and $C = (\langle F_1, \dots, F_k \rangle, \langle U_1, \dots, U_m \rangle, \langle O_1, \dots, O_n \rangle)$ be a cirquent. Then C^* is the game defined as follows, where Γ is an arbitrary run and Ω is any legal run of C^* .

(i) $\Gamma \in \mathbf{Lr}^{C^*}$ iff the following two conditions are satisfied:

- Every move of Γ looks like $a; \vec{u}.\alpha$, where α is some move, $a \in \{1, \dots, k\}$, and $\vec{u} = u_1, \dots, u_n$ is a sequence of n natural numbers such that, for every $j \in \{1, \dots, n\}$, we have $u_j = 0$ iff the overgroup O_j does not contain the oformula F_a .
- For every $a \in \{1, \dots, k\}$ and every sequence \vec{x} of n positive integers, $\Gamma^{[a;\vec{x}]} \in \mathbf{Lr}^{F_a^*}$.

(ii) $\mathbf{Wn}^{C^*} \langle \Omega \rangle = \top$ iff, for every $i \in \{1, \dots, m\}$ and every sequence \vec{x} of n positive integers, there is an $a \in \{1, \dots, k\}$ such that the undergroup U_i contains the oformula F_a and $\mathbf{Wn}^{F_a^*} \langle \Omega^{[a;\vec{x}]} \rangle = \top$.

Remark 3.3 Intuitively, any legal run Ω of C^* consists of parallel plays of countably infinite copies of each of the games F_a^* ($1 \leq a \leq k$). To every sequence \vec{x} of n positive integers corresponds a copy of F_a^* , and $\Omega^{[a;\vec{x}]}$ is the run played in that copy. We shall simply say **the copy** \vec{x} of F_a^* to mean the copy of F_a^* which corresponds to the sequence \vec{x} . Now, consider a given undergroup U_i . \top is the winner in U_i iff, for every sequence \vec{x} of n positive integers, there is an oformula F_a in U_i such that $\Omega^{[a;\vec{x}]}$ is won by \top . Finally, \top wins the overall game C^* iff it wins in all undergroups of C . In fact, overgroups can be seen as generalized \wedge s, with the only main difference that the former can be shared by several oformulas; undergroups can be seen as generalized disjunctions, with the only main difference that the former may have shared arguments with other undergroups.

We say that a cirquent C is **uniformly valid** iff there is an EPM \mathcal{M} , called a **uniform solution** of C , such that, for every interpretation $*$, \mathcal{M} wins C^* .

4 Main results

Lemma 4.1 *There is an effective function f from EPMs to EPMs such that, for every EPM \mathcal{M} , formula F and interpretation $*$, if \mathcal{M} wins λF^* , then $f(\mathcal{M})$ wins F^* .*

Proof. Our proof here almost literally follows the proof of Lemma 9.1 of [10]. It is known that affine logic proves $\lambda P \rightarrow P$. At the same time, according to Theorem 37 of [5], affine logic is sound with respect to uniform validity. So, the formula $\lambda P \rightarrow P$ is uniformly valid. This almost immediately implies that there is an EPM \mathcal{N}_0 such that \mathcal{N}_0 wins $\lambda F^* \rightarrow F^*$ for any formula F and interpretation $*$. Furthermore, by Proposition 21.3 of [1], there is an effective procedure that, for any pair $(\mathcal{N}, \mathcal{M})$ of EPMs, returns an EPM $h(\mathcal{N}, \mathcal{M})$ such that, for any static games A and B , if \mathcal{N} wins $A \rightarrow B$ and \mathcal{M} wins A , then $h(\mathcal{N}, \mathcal{M})$ wins B . So, let $f(\mathcal{M})$ be the function satisfying $f(\mathcal{M}) = h(\mathcal{N}_0, \mathcal{M})$. Then $f(\mathcal{M})$ wins F^* . ■

Lemma 4.2 *There is an effective function g from EPMs to EPMs such that, for every EPM \mathcal{M} , formula F and interpretation $*$, if \mathcal{M} wins $(F^\clubsuit)^*$, then $g(\mathcal{M})$ wins F^* .*

Proof. Again, it should be acknowledged that the present proof very closely follows the proof of Lemma 9.2 of [10], even though there are certain differences.

Every legal move of $(F^\clubsuit)^*$ looks like $1; u.\alpha$ for some positive integer u and move α , while the corresponding legal move of $(\lambda F)^*$ simply looks like $u.\alpha$, and vice versa. Consider an arbitrary EPM \mathcal{M} and an arbitrary interpretation $*$. Below we show the existence of an effective function f such that, if \mathcal{M} wins $(F^\clubsuit)^*$, then (the strategy) $f(\mathcal{M})$ wins $(\lambda F)^*$.

We construct an EPM $f(\mathcal{M})$ that plays $(\lambda F)^*$ by simulating and mimicking a play of $(F^\clubsuit)^*$ (called the **imaginary play**) by \mathcal{M} as follows. Throughout simulation, $f(\mathcal{M})$ grants permission whenever the simulated \mathcal{M} does so, and feeds its environment's response—in a slightly modified form described below—back to the simulated \mathcal{M} as the response of \mathcal{M} 's imaginary adversary (this detail of simulation will no longer be explicitly mentioned later in similar situations). Whenever the environment makes a move $u.\alpha$ for some positive integer u and move α , $f(\mathcal{M})$ translates it as the move $1; u.\alpha$ made by the imaginary adversary of \mathcal{M} , and “vice versa”: whenever the simulated \mathcal{M} makes a move $1; u.\alpha$ for some positive integer u and move α in the imaginary play of $(F^\clubsuit)^*$, $f(\mathcal{M})$ translates it as its own move $u.\alpha$ in the real play of $(\lambda F)^*$. The effect achieved by $f(\mathcal{M})$'s strategy can be summarized by saying that it synchronizes every copy of F^* in the real play of $(\lambda F)^*$ with the “same copy” of F^* in the imaginary play of $(F^\clubsuit)^*$.

Let Γ be an arbitrary run generated by $f(\mathcal{M})$, and Ω be the corresponding run in the imaginary play of $(F^\clubsuit)^*$ by \mathcal{M} . From our description of $f(\mathcal{M})$ it is clear that the latter never makes illegal moves unless its environment or the simulated \mathcal{M} does so first. Hence we may safely assume that Γ is a legal run of $(\lambda F)^*$ and Ω is a legal run of $(F^\clubsuit)^*$, for otherwise either Γ is a \perp -illegal run of $(\lambda F)^*$ and thus $f(\mathcal{M})$ is an automatic winner in $(\lambda F)^*$, or Ω is a \top -illegal run of $(F^\clubsuit)^*$ and thus \mathcal{M} does not win $(F^\clubsuit)^*$. Now, it is not hard to see that, for any positive integer x , we have $\Gamma^{x\cdot} = \Omega^{[1;x]}$. Therefore, $f(\mathcal{M})$ wins $(\lambda F)^*$ as long as \mathcal{M} wins $(F^\clubsuit)^*$.

Finally, in view of Lemma 4.1, the existence of function g satisfying the promise of the present lemma is obviously guaranteed. ■

A rule of **CL15**(λ) (other than Axiom) is said to be **uniform-constructively sound** iff there is an effective procedure that takes any instance (A, B) (i.e. a particular premise-conclusion pair) of the rule, any EPM \mathcal{M}_A and returns an EPM \mathcal{M}_B such that, for any interpretation $*$, whenever \mathcal{M}_A wins A^* , \mathcal{M}_B wins B^* . Axiom is uniform-constructively sound iff there is an effective procedure that takes any instance B of (the “conclusion” of) Axiom and returns a uniform solution \mathcal{M}_B of B .

Theorem 4.3 *All rules of **CL15**(λ) are uniform-constructively sound.*

Proof. In what follows, A is the premise of an arbitrary instance of a given rule of **CL15**(λ), and B is the corresponding conclusion, except the case of Axiom where we only have B . We will prove that each rule of **CL15**(λ) is uniform-constructively sound by showing that an EPM \mathcal{M}_B can be constructed effectively from an arbitrary EPM \mathcal{M}_A such that, for whatever interpretation $*$, whenever \mathcal{M}_A wins A^* , \mathcal{M}_B wins B^* . Since an interpretation $*$ is never relevant in such proofs, we may safely omit it, writing simply A instead of A^* to represent a game. Next, in all cases the assumption that \mathcal{M}_A wins A will be implicitly made, even though it should be pointed out that the construction of \mathcal{M}_B never depends on this assumption. Correspondingly, it will be assumed that \mathcal{M}_A never makes illegal moves. Further, as in the proof of Lemma 4.2, we shall always implicitly assume that \mathcal{M}_B 's adversary never makes illegal moves either. To summarize, when analyzing \mathcal{M}_B , \mathcal{M}_A and the games they play, we safely pretend that illegal runs never occur.

(1) Assume that B is an axiom with $2n$ oformulas. An EPM \mathcal{M}_B that wins B can be constructed as follows. It keeps granting permission. Whenever the environment makes a move $a; \vec{w}.\alpha$, where $1 \leq a \leq 2n$ and \vec{w} is a sequence of n natural numbers, \mathcal{M}_B responds by the move $b; \vec{w}.\alpha$, where $b = a + 1$ if a is odd, and $b = a - 1$ if a is even. Then, for any run Γ_B of B generated by \mathcal{M}_B and any sequence \vec{x} of n positive integers, we have $\Gamma_B^{[a;\vec{x}]} = \neg\Gamma_B^{[b;\vec{x}]}$. It is obvious that Γ_B is a \top -won run of B , so that \mathcal{M}_B wins B .

(2) Assume that B follows from A by Overgroup Exchange, where the i 'th ($i \geq 1$) and the $(i+1)$ 'th overgroups of A have been swapped when obtaining B from A . The EPM \mathcal{M}_B works by simulating and mimicking \mathcal{M}_A as follows. Let n be the number of overgroups of either cirquent, and a be a positive integer not exceeding the number of oformulas of either cirquent. For any move (by either player) $a; \vec{w}_1, u_1, u_2, \vec{w}_2.\alpha$ of the real play of B , where \vec{w}_1 and \vec{w}_2 are any sequences of $i - 1$ and $n - i - 1$ natural numbers, respectively, and u_1, u_2 are two natural numbers, \mathcal{M}_B translates it as the move $a; \vec{w}_1, u_2, u_1, \vec{w}_2.\alpha$ (by the same player) of the imaginary play of A , and vice versa, with all other moves

not reinterpreted. Let Γ_B be any run generated by \mathcal{M}_B , and Γ_A be the corresponding imaginary run generated by \mathcal{M}_A . It is obvious that, for any sequence \vec{x} of n positive integers, $\Gamma_B^{[a;\vec{x}]} = \Gamma_A^{[a;\vec{y}]}$, where \vec{y} is the result of swapping in \vec{x} the i 'th and $(i+1)$ 'th integers. Hence \mathcal{M}_B wins B (because \mathcal{M}_A wins A).

In the case of Oformula Exchange, a similar method can be used to construct \mathcal{M}_B , with the only difference that the reinterpreted objects are the occurrences of two adjacent oformulas rather than the occurrences of two adjacent overgroups.

As for Undergroup Exchange, its conclusion, as a game, is the same as its premise. So, the machine $\mathcal{M}_B = \mathcal{M}_A$ does the job.

In the subsequent clauses, as in the preceding one, without any further indication, Γ_B will stand for an arbitrary run of B generated by \mathcal{M}_B , and Γ_A will stand for the run of A generated by the simulated machine \mathcal{M}_A in the corresponding scenario.

(3) Assume B is obtained from A by Weakening. If no oformula of B was deleted when moving from B to A , then \mathcal{M}_B works exactly as \mathcal{M}_A does and succeeds, because every \top -won run of A is also a \top -won run of B (but not necessarily vice versa). If, when moving from B to A , an oformula F_a of B was deleted, then \mathcal{M}_B can be constructed as a machine that works by simulating and mimicking \mathcal{M}_A . What \mathcal{M}_B needs to do during its work is to ignore the moves within F_a , and play exactly as \mathcal{M}_A does in all other oformulas. Again, it is obvious that every \top -won run of A is also a \top -won run of B , which means that \mathcal{M}_B wins B as long as \mathcal{M}_A wins A .

(4) Since Exchange has already been proven to be uniform-constructively sound, in this and the remaining clauses of the present proof, we may safely assume that the oformulas and overgroups affected by a rule are at the end of the corresponding lists of objects of the corresponding cirquents.

Assume B follows from A by Contraction, and the contracted oformula $\text{Y}F$ is at the end of the list of oformulas of B . Let a be the number of oformulas of B , and let $b = a + 1$. Thus, the a 'th oformula of B is $\text{Y}F$, and the a 'th and b 'th oformulas of A are $\text{Y}F$ and $\text{Y}F$. Next, let n be the number of overgroups in either cirquent. As always, we let \mathcal{M}_B be an EPM that works by simulating and mimicking \mathcal{M}_A . Namely, let \vec{w} be any sequence of n natural numbers. If the moves take place within the oformulas other than $\text{Y}F$, then nothing should be reinterpreted. If the moves take place in $\text{Y}F$, then we have:

- For any move $a; \vec{w}.u.\alpha$ (by either player) in the real play of B , where $u = 2k - 1$ for some $k \in \{1, 2, 3, \dots\}$, \mathcal{M}_B translates it as the move $a; \vec{w}.k.\alpha$ (by the same player) of the imaginary play of A , and vice versa.
- For any move $a; \vec{w}.v.\alpha$ (by either player) in the real play of B , where $v = 2m$ for some $m \in \{1, 2, 3, \dots\}$, \mathcal{M}_B translates it as the move $b; \vec{w}.m.\alpha$ (by the same player) of the imaginary play of A , and vice versa.

Below we will show that \mathcal{M}_B wins B , i.e., \mathcal{M}_B is the winner in every undergroup of B . Let U_i^B be any i 'th undergroup of B and U_i^A be the corresponding i 'th undergroup of A , and let \vec{x} be any sequence of n positive integers. Since \mathcal{M}_A wins A , U_i^A is won by \mathcal{M}_A . So, for the sequence \vec{x} , there is an oformula F_j ($1 \leq j \leq b$) in U_i^A such that $\Gamma_A^{[j;\vec{x}]}$ is a \top -won run of F_j . Next, if such F_j is not one of the two contracted oformulas $\text{Y}F$ and $\text{Y}F$, then, for \vec{x} , the corresponding oformula F_j of B is also won by \mathcal{M}_B , i.e. $\Gamma_B^{[j;\vec{x}]}$ is a \top -won run of F_j , because \mathcal{M}_B plays in the copy \vec{x} of F_j exactly as \mathcal{M}_A does. This means that U_i^B is won by \mathcal{M}_B . If such F_j is one of the two contracted oformulas $\text{Y}F$ and $\text{Y}F$, below let us assume that F_j is the left $\text{Y}F$, with the case of the right $\text{Y}F$ being similar. Then there is a positive integer w such that the w 'th component F of the copy \vec{x} of the left $\text{Y}F$ is won by \mathcal{M}_A , i.e. $(\Gamma_A^{[j;\vec{x}]})^w$ is a \top -won run of F . But, according to the above description, \mathcal{M}_B plays in the $(2w-1)$ 'th component F of the copy \vec{x} of $\text{Y}F$ in B exactly as \mathcal{M}_A plays in the w 'th component F of the copy \vec{x} of the left $\text{Y}F$ in A , i.e. $(\Gamma_B^{[j;\vec{x}]})^{(2w-1)} = (\Gamma_A^{[j;\vec{x}]})^w$. Therefore, $(\Gamma_B^{[j;\vec{x}]})^{(2w-1)}$ is a \top -won run of F , which means that $\Gamma_B^{[j;\vec{x}]}$ is a \top -won run of $\text{Y}F$ in B , and hence the $\text{Y}F$ -containing undergroup U_i^B is won by \mathcal{M}_B .

Remark: In the remaining clauses, just as in the preceding one, when talking about playing, winning, etc. in A (resp. B) or any of its components, it is to be understood in the context of Γ_A (resp. Γ_B). Furthermore, if A and B have the same number n of overgroups, then the context will additionally include some arbitrary but fixed sequence \vec{x} of n positive integers.

(5) Undergroup Duplication does not modify the game associated with the cirquent, so we only need to consider Overgroup Duplication.

Assume B is obtained from A by Overgroup Duplication. We assume that the duplicated overgroup is at the end of the list of overgroups of A . Let $n+1$ be the number of overgroups of A . Thus, every legal move of A (resp. B) looks like $a; \vec{w}, u.\alpha$ (resp. $a; \vec{w}, u_1, u_2.\alpha$), where a is a positive integer not exceeding the number of oformulas of A , \vec{w} is a sequence of n natural numbers, and u, u_1, u_2 are natural numbers.

Let f be some standard 1-to-1 correspondence from the set of all pairs of positive integers to the set of all positive integers. As before, \mathcal{M}_B works by simulating \mathcal{M}_A . Whenever \mathcal{M}_A makes a move $a; \vec{w}, 0.\alpha$ in A , \mathcal{M}_B makes the move $a; \vec{w}, 0, 0.\alpha$ in the real play of B , and vice versa. Whenever \mathcal{M}_A makes the move $a; \vec{w}, u.\alpha$ in A for some positive integer u , \mathcal{M}_B makes the move $a; \vec{w}, u_1, u_2.\alpha$ in B , where u_1, u_2 are integers with $f(u_1, u_2) = u$, and vice versa. Note that \mathcal{M}_A 's (legally) making a move $a; \vec{w}, 0.\alpha$ means that the a 'th oformula F_a of A is not contained in the $(n+1)$ 'th overgroup O_{n+1} that was duplicated when moving from A to B , which, in turn, means that the corresponding F_a of B is contained in neither the $(n+1)$ 'th overgroup O'_{n+1} nor the $(n+2)$ 'th overgroup O'_{n+2} of B . Similarly, if \mathcal{M}_A makes a move $a; \vec{w}, u.\alpha$ for some positive integer u , then F_a is contained in O_{n+1} of A , and hence the corresponding F_a of B is contained in both O'_{n+1} and O'_{n+2} of B , with the case of F_a being contained in O'_{n+1} but not in O'_{n+2} (or in O'_{n+2} but not in O'_{n+1}) being impossible.

For every oformula F_a of either cirquent, every sequence \vec{y} of n positive integers and any positive integers x_1 and x_2 , we have $\Gamma_B^{[a; \vec{y}, x_1, x_2]} = \Gamma_A^{[a; \vec{y}, x]}$, where $x = f(x_1, x_2)$. So it is obvious that \mathcal{M}_B wins B as long as \mathcal{M}_A wins A .

(6) Assume B follows from A by Merging. Let us assume that A has $n+2$ overgroups, and B is the result of merging in A the two adjacent overgroups O_{n+1} and O_{n+2} . Then every legal move of A (resp. B) looks like $a; \vec{w}, u_1, u_2.\alpha$ (resp. $a; \vec{w}, u.\alpha$), where a is a positive integer not exceeding the number of oformulas in either cirquent, \vec{w} is a sequence of n natural numbers, and u, u_1, u_2 are natural numbers. The EPM \mathcal{M}_B works as follows.

If the a 'th oformula of A is neither in O_{n+1} nor in O_{n+2} , then \mathcal{M}_B interprets every move $a; \vec{w}, 0, 0.\alpha$ made by \mathcal{M}_A in the imaginary play of A as the move $a; \vec{w}, 0.\alpha$ in the real play of B , and vice versa.

If the a 'th oformula of A is in O_{n+1} but not in O_{n+2} , \mathcal{M}_B interprets every move $a; \vec{w}, v, 0.\alpha$ (v is a positive integer) made by \mathcal{M}_A in the imaginary play of A as the move $a; \vec{w}, v.\alpha$ that \mathcal{M}_B itself should make in the real play of B , and vice versa. Namely, \mathcal{M}_B interprets every move $a; \vec{w}, v.\alpha$ by its environment in the real play of B as the move $a; \vec{w}, v, 0.\alpha$ by \mathcal{M}_A 's adversary in the imaginary play of A .

The case of the a 'th oformula of A being in O_{n+2} but not in O_{n+1} is similar.

Now assume that the a 'th oformula of A is in both O_{n+1} and O_{n+2} . \mathcal{M}_B interprets every move $a; \vec{w}, v_1, v_2.\alpha$ by \mathcal{M}_A in the imaginary play of A as the move $a; \vec{w}, v.\alpha$ in the real play of B , where v_1, v_2, v are positive integers such that $v = f(v_1, v_2)$, with f here standing for the pairing function explained in the preceding clause of this proof.

For every oformula F_a of either cirquent, every sequence \vec{y} of n positive integers and any positive integer x , we have $\Gamma_B^{[a; \vec{y}, x]} = \Gamma_A^{[a; \vec{y}, x_1, x_2]}$, where x_1, x_2 are positive integers satisfying that $x_1 = x$ (when F_a is contained in O_{n+1} but not O_{n+2}), or $x_2 = x$ (when F_a is contained in O_{n+2} but not O_{n+1}), or $f(x_1, x_2) = x$ (when F_a is contained in both O_{n+1} and O_{n+2} , or is contained in neither of them). So it is obvious that \mathcal{M}_B wins B as long as \mathcal{M}_A wins A .

(7) In this and the remaining clauses of the present proof, we will limit our descriptions to what moves \mathcal{M}_B needs to properly reinterpret and how, with any unmentioned sorts of moves implicitly assumed to remain unchanged.

Assume B is obtained from A by Disjunction Introduction. Let us assume that the last (a 'th) oformula of B is $E \vee F$, and the last two (a 'th and b 'th, where $b = a+1$) oformulas of A are E and F .

We let \mathcal{M}_B reinterpret every move $a; \vec{w}.\alpha$ (resp. $b; \vec{w}.\alpha$) by either player in the imaginary play of A as the move $a; \vec{w}.1.\alpha$ (resp. $a; \vec{w}.2.\alpha$) by the same player in the real play of B , and vice versa.

Consider any undergroup U_i^B of B , and let U_i^A be the corresponding undergroup of A . As before, \mathcal{M}_A 's winning A means that U_i^A is won by \mathcal{M}_A , which, in turn, means that there is an oformula G in U_i^A that is won by \mathcal{M}_A . If G is neither E nor F , then the oformula G of B is also won by \mathcal{M}_B , because \mathcal{M}_B plays in G exactly as \mathcal{M}_A does. Hence U_i^B is won by \mathcal{M}_B . If G is E , then its being \top -won means that \mathcal{M}_B wins the E component of $E \vee F$, because \mathcal{M}_B plays in the E component of $E \vee F$ exactly as \mathcal{M}_A plays in E . Therefore, $E \vee F$ is won by \mathcal{M}_B , and hence so is the $E \vee F$ -containing undergroup U_i^B . The case of G being F is similar.

(8) Assume B follows from A by Conjunction Introduction. We also assume that the last (a 'th) oformula of B is $E \wedge F$, and the last two (a 'th and b 'th, where $b = a + 1$) oformulas of A are E and F . As the case of Disjunction Introduction, \mathcal{M}_B reinterprets every move $a; \vec{w}.\alpha$ (resp. $b; \vec{w}.\alpha$) by either player in the imaginary play of A as the move $a; \vec{w}.1.\alpha$ (resp. $a; \vec{w}.2.\alpha$) by the same player in the real play of B , and vice versa.

Let U_i be any undergroup of B . If U_i does not contain $E \wedge F$, then the corresponding undergroup V_i of A contains neither E nor F . In this case, U_i is won by \mathcal{M}_B for the same reason as in the preceding clause. If U_i contains $E \wedge F$, then there are two undergroups V_i^E, V_i^F of A corresponding to U_i , where V_i^E contains E (but not F), and V_i^F contains F (but not E), with all other ($\neq E \wedge F$) oformulas of U_i contained by both V_i^E and V_i^F . Of course, both V_i^E and V_i^F are won by \mathcal{M}_A because \mathcal{M}_A wins the overall game A . This means that there is an oformula G_1 (resp. G_2) in V_i^E (resp. V_i^F) such that \mathcal{M}_A wins it. If at least one oformua $G \in \{G_1, G_2\}$ is neither E nor F , then the corresponding oformula G of B is won by \mathcal{M}_B , because \mathcal{M}_B plays in G exactly as \mathcal{M}_A does. Hence the G -containing undergroup U_i of B is won by \mathcal{M}_B . If G_1 is E and G_2 is F , then \mathcal{M}_A winning them means that \mathcal{M}_B wins both the E and the F components of $E \wedge F$, because \mathcal{M}_B plays in the E (resp. F) component of $E \wedge F$ exactly as \mathcal{M}_A does in E (resp. F). Hence $E \wedge F$ is won by \mathcal{M}_B , and hence so is the $E \wedge F$ -containing undergroup U_i .

(9) Assume B is obtained from A by Recurrence Introduction. Namely, the last (a 'th) oformula of B is λF , and the last (a 'th) oformula of A is F . We further assume that the number of overgroups of B is n , and thus the number of overgroups of A is $n + 1$. In what follows, \vec{w} is any sequence of n natural numbers, and b is a positive integer not exceeding the number of oformulas of either cirquent. If $b \neq a$, then \mathcal{M}_B simply reinterprets every move $b; \vec{w}, 0.\alpha$ by either player in the imaginary play of A as the move $b; \vec{w}.\alpha$ by the same player in the real play of B , and vice versa. If $b = a$, then \mathcal{M}_B reinterprets, for any positive integer u , every move $a; \vec{w}, u.\alpha$ by either player in the imaginary play of A as the move $a; \vec{w}.u.\alpha$ by the same player in the real play of B , and vice versa.

Consider any undergroup U_i^B of B . Let $\vec{x} = x_1, \dots, x_n$ be any sequence of n positive integers. \mathcal{M}_A 's winning A means that Γ_A is a \top -won run of A and that the corresponding undergroup U_i^A of A is won by \mathcal{M}_A . Then, for any sequence $\vec{y} = x_1, \dots, x_n, x$, where x is any positive integer, there is an oformula F_b in U_i^A such that $\Gamma_A^{[b;\vec{y}]}$ is a \top -won run of F_b . If such F_b is not the a 'th oformula F , then, in the context of \vec{x} , the oformula F_b of B is also won by \mathcal{M}_B , i.e. $\Gamma_B^{[b;\vec{x}]}$ is a \top -won run of F_b , because \mathcal{M}_B plays in the copy \vec{x} of F_b in B exactly as \mathcal{M}_A does in the copy \vec{y} of F_b in A . Hence U_i^B is won by \mathcal{M}_B . If F_b is the a 'th oformula F , then, in the context of \vec{x} , the corresponding oformula λF of B is won by \mathcal{M}_B as well, i.e. $\Gamma_B^{[a;\vec{x}]}$ is a \top -won run of λF . This is so because \mathcal{M}_B plays in the x 'th component F of the copy \vec{x} of λF exactly as \mathcal{M}_A does in the copy \vec{y} of F in A . Namely, $(\Gamma_B^{[a;\vec{x}]})^{x.} = \Gamma_A^{[a;\vec{y}]}$. Since $\Gamma_A^{[a;\vec{y}]}$ is a \top -won run of F , so is $(\Gamma_B^{[a;\vec{x}]})^{x.}$. Further, due to the arbitrariness of x , $\Gamma_B^{[a;\vec{x}]}$ is a \top -won run of λF . Therefore, the λF -containing undergroup U_i^B is won by \mathcal{M}_B .

(10) Finally, assume that B is obtained from A by Corecurrence Introduction. Let us assume that the last (a 'th) oformula of B is $\forall F$, and the last (a 'th) oformula of A is F . And assume that n ($n \geq 0$) is the number of the *new* overgroups U_j in which the a 'th oformula F was included when moving from B to A . Let us further assume that all of such n overgroups are at the end of the list of overgroups of either cirquent. In what follows, let \vec{w} be any sequence of m natural numbers, where m is the total number of overgroups of either cirquent minus n . We construct the EPM \mathcal{M}_B as follows.

Let f be some standard injective function from the set of n -tuples (u_1, \dots, u_n) of positive integers onto the set of positive integers u . In its simulation routine, \mathcal{M}_B reinterprets every move $a; \vec{w}, u_1, \dots, u_n. \alpha$ made by \mathcal{M}_A in the imaginary play of A as the move $a; \vec{w}, 0, \dots, 0. u. \alpha$ (n occurrences of 0 after \vec{w}) in the real play of B , where $u = f(u_1, \dots, u_n)$. Whenever the environment makes a move $a; \vec{w}, 0, \dots, 0. v. \beta$ (also n occurrences of 0 after \vec{w}) for some positive integer v in the real play of B , if there is no n -tuple (u_1, \dots, u_n) such that $v = f(u_1, \dots, u_n)$, then \mathcal{M}_B simply ignores it; if $v = f(u_1, \dots, u_n)$, then \mathcal{M}_B translates it as the move $a; \vec{w}, u_1, \dots, u_n. \beta$ by \mathcal{M}_A 's adversary in the imaginary play. Note that the above routine works as well in the case of $n = 0$. Simply, $f() = c$ for some fixed positive integer c , \mathcal{M}_B reinterprets every move $a; \vec{w}. \alpha$ made by \mathcal{M}_A in A as the move $a; \vec{w}. c. \alpha$ in B , and whenever the environment makes a move $a; \vec{w}. v. \beta$ in B , if $v \neq c$, \mathcal{M}_B ignores it, and if $v = c$, \mathcal{M}_B translates it as the move $a; \vec{w}. \beta$ by \mathcal{M}_A 's adversary in the imaginary play of A .

As usual, consider any undergroup U_i^B of B , and let $\vec{x} = \vec{y}, x_1, \dots, x_n$ be any sequence of $(m + n)$ positive integers, where \vec{y} is any sequence of m positive integers. Then the corresponding undergroup U_i^A of A is won by \mathcal{M}_A , which, in turn, means that there is an oformula F_b ($1 \leq b \leq a$) in U_i^A such that \mathcal{M}_A wins it. If such F_b is not the a 'th oformula F , then the corresponding oformula F_b of B is also won by \mathcal{M}_B , because \mathcal{M}_B plays in F_b of B exactly as \mathcal{M}_A does in F_b of A . Therefore, the F_b -containing undergroup U_i^B is won by \mathcal{M}_B . If F_b is the a 'th oformula F , then the corresponding oformula γF of B is won by \mathcal{M}_B as well. This is so because \mathcal{M}_B plays in at least one component F of γF in B exactly as \mathcal{M}_A does in F of A . Precisely, we have $(\Gamma_B^{[a; \vec{y}, x_1, \dots, x_n]})_x = \Gamma_A^{[a; \vec{y}, x_1, \dots, x_n]}$, where $x = f(x_1, \dots, x_n)$. Thus the γF -containing undergroup U_i^B is won by \mathcal{M}_B . ■

Theorem 4.4 *Every cirquent provable in $\mathbf{CL15}(\lambda)$ is uniformly valid.*

Furthermore, there is an effective procedure that takes an arbitrary $\mathbf{CL15}(\lambda)$ -proof of an arbitrary cirquent C and constructs a uniform solution of C .

Proof. Immediately from Theorem 4.3 by induction on the lengths of $\mathbf{CL15}(\lambda)$ -proofs. ■

Theorem 4.5 *For any formula F , if $\mathbf{CL15}(\lambda) \vdash F$, then F is uniformly valid.*

Furthermore, there is an effective procedure which takes any $\mathbf{CL15}(\lambda)$ -proof of any formula F and constructs a uniform solution of F .

Proof. Immediately from Theorem 4.4 and Lemma 4.2. ■

Below, a **uniformly valid** $(\neg, \wedge, \vee, \lambda, \gamma)$ -**principle** means the result of replacing every occurrence of the operator λ (resp. γ) by the symbol ! (resp. ?) in some uniformly valid $(\neg, \wedge, \vee, \lambda, \gamma)$ -formula. Similarly, a **uniformly valid** $(\neg, \wedge, \vee, \circ, \dagger)$ -**principle** means the result of replacing every occurrence of the operator \circ (resp. \dagger) by the symbol ! (resp. ?) in some uniformly valid $(\neg, \wedge, \vee, \circ, \dagger)$ -formula. The reason for introducing these technical concepts is merely to make it possible to directly compare the otherwise syntactically nonidentical $(\neg, \wedge, \vee, \lambda, \gamma)$ -formulas with $(\neg, \wedge, \vee, \circ, \dagger)$ -formulas.

Theorem 4.6 *The set of uniformly valid $(\neg, \wedge, \vee, \lambda, \gamma)$ -principles is a proper superset of the set of uniformly valid $(\neg, \wedge, \vee, \circ, \dagger)$ -principles.*

Proof. The fact that the set of uniformly valid $(\neg, \wedge, \vee, \lambda, \gamma)$ -principles is a *superset* of the set of uniformly valid $(\neg, \wedge, \vee, \circ, \dagger)$ -principles is immediate from Theorems 2.1 and 4.5. Furthermore, the former set is in fact a *proper* superset of the latter set because, as proven in [8], the formula $P \wedge \lambda(P \rightarrow P \wedge P) \rightarrow \lambda P$ is uniformly valid while its counterpart $P \wedge \circ(P \rightarrow P \wedge P) \rightarrow \circ P$ is not. ■

5 A secondary result

Japaridze [5, 6] claimed that \Diamond is strictly stronger than \wedge (and thus \Diamond is strictly weaker than \forall) in the sense that the formula $\Diamond P \rightarrow \wedge P$ is uniformly valid while its converse $\wedge P \rightarrow \Diamond P$ is not. The first part of this claim was proven in [3], but the second part has never been verified. In order to make our investigation of the relationship between the two sorts of recurrences more comprehensive, below we provide such a verification.

Theorem 5.1 *The formula $\wedge P \rightarrow \Diamond P$ is not uniformly valid.*

Proof. Let \mathcal{M} be an arbitrary EPM, i.e. strategy of the machine (\top). Below we construct a counterstrategy \mathcal{C} such that, when the environment (\perp) follows it, \mathcal{M} loses $\wedge P \rightarrow \Diamond P$ with P interpreted as a certain enumeration game. Here, an **enumeration game** ([8]) is a game where any natural number, identified with its decimal representation, is a legal move by either player at any time (and there are no other legal moves). It should be noted that, as shown in [11], every enumeration game is static, and hence is a legitimate value of an interpretation $*$ on any atom. Hence, due to the arbitrariness of \mathcal{M} , $\wedge P \rightarrow \Diamond P$ (i.e. $\forall \neg P \vee \Diamond P$) is not uniformly valid.

Since P is going to be interpreted as an enumeration game and its legal moves are known even before we actually define that interpretation, in certain contexts we may identify formulas with games without creating any confusion. The work of \mathcal{C} consists in repeating the following interactive routine over and over again (infinitely many times), where i is the number of the iteration. In our description below, a *fresh number* means a natural number that has not yet been chosen in the play by either player as a move in any thread/copy of P .

LOOP(i): Whenever permission is granted by the machine \mathcal{M} , make the move $2.w.u$, where u is a fresh number and w is the i th finite bitstring of the lexicographic list of all finite bitstrings.

Consider the run Δ generated by \mathcal{M} in the scenario when its adversary follows the above counter-strategy. Let $\Omega = \Delta^1$ and $\Gamma = \Delta^2$. That is, Ω is the (sub)run that took place in the $\forall \neg P$ component, and Γ is the (sub)run that took place in the $\Diamond P$ component. From some analysis of the work of LOOP, details of which are left to the reader, one can see that $\Gamma^{\leq x_1} \neq \Gamma^{\leq x_2}$ for any two different infinite bitstrings x_1 and x_2 . Hence, as there are uncountably many infinite bitstrings while only countably many positive integers, there is an infinite bitstring y such that, for every positive integer v , $\Omega^v \neq \neg \Gamma^{\leq y}$. Fix this y .

Now we select an interpretation $*$ that interprets P as the enumeration game such that, for any legal run Θ of the game P , $Wn^P\langle\Theta\rangle = \perp$ iff $\Theta = \Gamma^{\leq y}$. We claim that \mathcal{M} loses the overall game under this interpretation. First, it is obvious that \mathcal{M} loses the game P in the thread y , which means that it loses the $\Diamond P$ component. Next, \mathcal{M} also loses the $\forall \neg P$ component because it loses every component $\neg P$ of $\forall \neg P$. This is so because the run that took place in any component $\neg P$ of $\forall \neg P$ is won by \top iff it is $\neg \Gamma^{\leq y}$, which, however, is impossible (due to the above analysis). ■

An alternative albeit non-constructive and less direct proof of Theorem 5.1 would rely on Theorem 4.6. Namely, one could show that, if $\wedge P \rightarrow \Diamond P$ was uniformly valid and hence (in view of the already known fact of the uniform validity of the converse of this formula) $\Diamond P$ and $\wedge P$ were “logically equivalent”, then they would induce identical logics, in the precise sense that the set of uniformly valid $(\neg, \wedge, \vee, \wedge, \forall)$ -principles would coincide with the set of uniformly valid $(\neg, \wedge, \vee, \Diamond, \exists)$ -principles, contrary to what Theorem 4.6 asserts.

References

- [1] G. Japaridze. *Introduction to computability logic*. **Annals of Pure and Applied Logic** 123 (2003), pp. 1-99.
- [2] G. Japaridze. *Introduction to cirquent calculus and abstract resource semantics*. **Journal of Logic and Computation** 16 (2006), pp. 489-532.

- [3] G. Japaridze. *The logic of interactive Turing reduction*. **Journal of Symbolic Logic** 72 (2007), No.1, pp. 243-276.
- [4] G. Japaridze. *Cirquent calculus deepened*. **Journal of Logic and Computation** 18 (2008), pp. 983-1028.
- [5] G. Japaridze. *In the beginning was game semantics*. In: **Games: Unifying Logic, Language and Philosophy**. O. Majer, A.-V. Pietarinen and T. Tulenheimo, eds. Springer 2009, pp. 249-350.
- [6] G. Japaridze. *Many concepts and two logics of algorithmic reduction*. **Studia Logica** 91 (2009), pp. 1-24.
- [7] G. Japaridze. *From formulas to cirquents in computability logic*. **Logical Methods in Computer Science** 7 (2011), Issue 2, Paper 1, pp. 1-55.
- [8] G. Japaridze. *Separating the basic logics of the basic recurrences*. Manuscript (2010). Available online at <http://arxiv.org/abs/1007.1324>
- [9] G. Japaridze. *A new face of the branching recurrence of computability logic*. Manuscript (2011). Available online at <http://arxiv.org/abs/1102.1054>
- [10] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part I*. Manuscript (2011). Available online at <http://arxiv.org/abs/1105.3853>
- [11] G. Japaridze. *The taming of recurrences in computability logic through cirquent calculus, Part II*. Manuscript (2011). Available online at <http://arxiv.org/abs/1106.3705>
- [12] I. Mezhirov, N. Vereshchagin. *On abstract resource semantics and computability logic*. **Journal of Computer and System Sciences** 76 (2010), pp. 356-372.
- [13] Wen-yan Xu, San-yang Liu. *Deduction theorem for symmetric cirquent calculus*. **Quantitative Logic and Soft Computing 2010, Advances in Intelligent and Soft Computing** 82 (2010), pp. 121-126.
- [14] Wen-yan Xu, San-yang Liu. *Soundness and completeness of the cirquent calculus system CL6 for computability logic*. Manuscript (2011). Available online at <http://arxiv.org/abs/1106.2272>